

TDK-dolgozat

Dénes Botond

Vízközeli hulladéklerakók megbízható detektálása multispektrális műholdfelvételek segítségével

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI TANSZÉK



Szerző:

Dénes Botond

programtervező informatikus MSc

2. évfolyam

Témavezető:

Cserép Máté

egyetemi tanársegéd

Budapest, 2024

Tartalomjegyzék

1. Bevezetés	3
1.0.1. A kutatólabor eddigi eredményei	4
2. Kapcsolódó kutatások	5
2.0.1. Hulladékdetektálási módszerek	5
2.0.2. PlanetScope	6
3. Elemzés és tervezés	7
3.0.1. Kutatási cél	7
3.0.2. Műhold specifikációk	7
3.0.3. Használt indexek	7
4. Betanítás	8
4.0.1. Tanítóadatok	8
4.0.2. Tanítási paraméterek	9
4.0.3. Főkomponens analízis(PCA)	9
4.0.4. Nyári és téli adatokra való lebontás	10
5. Tesztelés	12
5.0.1. Mérési módszerek	12
6. Megvalósítás és alkalmazás	13
6.0.1. A meglevő alkalmazások bővítése	13
6.0.2. A Tisza-Tisza alkalmazás	13
6.0.3. Közös metszet	17
7. Összefoglalás és eredmények	18
Köszönetnyilvánítás	19

A. Szimulációs eredmények	20
Irodalomjegyzék	21
Ábrajegyzék	23
Táblázatjegyzék	24
Algoritmusjegyzék	25
Forráskódjegyzék	26

1. fejezet

Bevezetés

szinkronizálni az
absztrakttal

A hulladékszennyezés komoly problémát jelent a természet számára [1]. Emiatt számos szervezet mozdul abba az irányba, hogy tisztábbá tegye a bolygónkat. Egy ilyen szervezet a PET Kupa, akik folyómenti hulladékgyűjtéssel foglalkoznak elsősorban Magyarországon, de figyelmük kiterjed a szomszédos országokra is. Az egyik nagy kihívás a szemétgyűjtésben a hulladékkal szennyezett területeknek a hatékony megtalálása. Sok erőforrást igényel a hulladéklerakók megtalálása a folyók mentén, hiszen sokszor járművel kell valakinek végig haladnia egy hosszabb területen, azért, hogy felmérje, hogy hol van hulladék. A folyók árterén elhelyezett hulladékok még nagyobb problémát jelentenek, hiszen dagály idejében a hulladékot elmossa a víz és ez a folyó további szakaszaira lesz szétszórva miközben nagy károkat okoz a folyó élővilágának, illetve szennyezi a folyóvizet [2, 3]. Emiatt szükségünk van olyan eszközökre, melyekkel hamar lehet detektálni a szennyezett területeket, hogy ezeket minél hatékonyabban meglehessen tisztítani. Ennek fényében a PET Kupa felkereste az egyetemünket azzal a kéréssel, hogy olyan eszközöket fejlesszünk le, melyek automatikusan képesek lesznek hulladékot detektálni a folyók mentén.

Hivatkozni a
weboldalra

A dolgozatomban bemutatok egy Random Forest modellt [4], mely a kutatólaborban már lefejlesztett modellre épül [5]. A bemutatott modell javít a korábbi megoldás pontosságán, illetve nagyobb megbízhatósággal találja meg a hulladékot a folyókon és a folyók mentén. A modell eredményei integrálásra kerülnek a Tiszta Tisza webalkalmazásba, ahol több napon keresztül történő detektálás eredménye lesz összesítve és megjelenítve a felhasználók számára. Ezen felül a dolgozatban tárgyalni fogok más kutatást is, mely a hulladékdetektálás problémájával foglalkozik. Ezen kívül kiegészítem a kutatólabor meglevő szoftveres eszközeit annak érdekében, hogy a laborban zajló

munka gördülékenyebb legyen.

A kutatás hozzáadott terméke egy olyan adathalmaz, mely alkalmas más hulladékdetektálási modellek betanítására is. Az adathalmaz elsősorban szárazföldi romániai hulladéklerakókról készített PlanetScope műholdfelvételeket tartalmaz, melyek kézzel voltak annotálva. Az adatok georeferálva vannak, így ezeket könnyen meg lehet vizsgálni, illetve ki lehet egészíteni.

Továbbá bemutatom, hogy milyen módszerekkel próbáltam tovább javítani a modell eredményein. Ilyen módszer például a főkomponens analízis, a képnormalizálás, illetve az évszakokra bontás.

1.0.1. A kutatólabor eddigi eredményei

A térinformatikai kutatólaborban már fejlesztésre került egy szerveralkalmazás, mely minden nap a Planet-ről letölti a legfrissebb felvételeket a vizsgált területekről, és lefuttatja ezeken a képeken az akkori modellt. Ezen felül készült egy webalkalmazás is, ami erről a szerverről letölti az eredményeket, és megjeleníti ezeket, összehasonlításra. A kutatólabor rendelkezik egy asztali alkalmazással is, mellyel hatékonyan elő lehet állítani tanítóadatokat. A kutatásom elősegítéséhez ezeket az alkalmazásokat használtam, illetve bővítettem a 6.0.1 fejezetben leírtak szerint.

2. fejezet

Kapcsolódó kutatások

2.0.1. Hulladékdetektálási módszerek

Számos kutatás készült már a hulladékdetektálás témájában. Az én kutatásom a Térinformatikai kutatólabor munkájára épül [5], ahol egy Random Forest modell került betanításra, hulladékdetektálás céljából. A kutatásban PlanetScope és Sentinel-2 műholdfelvételeket használtak. Ez a cikk rakta le az alapjait az én kutatásomnak is, melyben ezeken az eredményeken javítok. A dolgozatomban csak a PlanetScope felvételek kerülnek felhasználásra, mivel a magasabb felbontású felvételek könnyebben lehetővé teszik a tanítóadatok előállítását.

[6]-ben Sentinel-2 műholdfelvételeken tanítottak be egy Random Forest modell-t egy indonéziai folyón. A cikkben bevezetik az "Adjusted Plastic Index"-et, mellyel a vegetáció, föld és épületek közötti zajt csökkentik. Ennek az indexnek a kiszámításához a Sentinel-2 műhold piros, közeli infravörös (NIR), illetve rövid hullámhosszú közeli infravörös (SWIR) sávokat használták fel. Validációnak Pleiades műholdképeket és drónfelvételeket klasszifikáltak Mahalanobis távolság gépi tanulási módszerrel. A módszer növényzetten és vízen rendre 88%, illetve 85% pontosságot ért el és épületeken, törmeléken és földön rendre 62%, 53%, illetve 21% pontossággal tudta a hulladékot detektálni. A cikk szerint az utóbbi három adattípuson azért visszafogottabbak az arányok, mert a spektrális értékei az épületeknek, a földnek és a törmeléknek nagyon hasonlítanak.

[7]-ben Spectral Angle Mapping módszert alkalmaztak multispektrális drónfelvételeken. A célja a kutatásnak az volt, hogy a tengerparton kimosott hulladékot detektálják és klasszifikálják. A módszer alkalmazásához referencia értékeket állítottak elő azzal, hogy elhelyeztek különböző anyagokból álló hulladékot a homokba, és ezekről drónfelvételt készítettek. Ezzel a módszerrel képesek voltak detektálni és klasszifikálni

nem csak homok fölött található hulladékot, hanem félig elásott hulladékot is. A 472 kézzel előállított tesztadatból volt a 268 True Positive(57% összesen), 96 volt a False Positive és 204 volt a False Negative.

[8]-ben mezőgazdasági hulladékdetektálásra használtak egy Support Vector Machine modellt, Landsat 8 műholdfelvételeken. A szenzor Kék, Zöld, Piros, NIR, SWIR 1, SWIR 2 és CIRRUS sávját használták a tanítóadatok és tesztadatok előállítására. Ezután véletlenszerűen szétválasztották az adatokat tanítóadatokra és tesztadatokra. A következő osztályokra bontották az adatokat: Hálók, műanyag takarók, föld, növényzet, gyümölcsöskert, olajfás kert, város, fa, fás föld. A modell a tesztadatokat összességében 94%-os pontossággal tudta klasszifikálni, ahol a legrosszabb arányokat az olajfás kert érte el 77.78%-os pontossággal.

2.0.2. PlanetScope

3. fejezet

Elemzés és tervezés

3.0.1. Kutatási cél

A cél az, hogy a kutatás során szerzett modell megbízhatóan detektáljon hulladéklerakókat. Ehhez a false positive arányok minél kisebbek kell legyenek, míg a true positive arányok minél nagyobbak. Ugyanakkor nem jelent ugyanakkora problémát egy false negative, mint egy false positive, mivel a false positive eredmények fölöslegesen rossz irányba küldhetik a folyómentő csapatot. A kutatólabor 2023-as cikkjében bemutatott modell (továbbiakban meglevő modell) egyik problémája a nagy false positive arányok voltak. A modell a pusztazámori hulladéklerakóról, illetve a kiskörei víztárolóról szerzett adatokkal volt betanítva. Ezért érdemes első körben egy nagyobb adathalmazzal betanítani a modellt.

3.0.2. Műhold specifikációk

Az új Random Forest modell a PlanetScope műholdakra lesz specializálva, azon belül is a legújabb PSB.SD szenzorokra[9]. A modell számára elérhető lesz a Vörös, Kék, Zöld, és a közeli infravörös (NIR) sáv. A PlanetScope műholdak körülbelül 3 méter/pixel felbontással rendelkeznek [10].

3.0.3. Használt indexek

A kutatás során felhasználok a kutatólaborban már számolt indexeket. Pontosabban a Plastic Index (PI), Normalized Difference Water Index (NDWI), Normalized Difference Vegetation Index (NDVI), Reversed Normalized Difference Vegetation Index (RNDVI), Simple Ratio (SR) indexek kerülnek használatra.

4. fejezet

Betanítás

4.0.1. Tanítóadatok

A betanításhoz 29 romániai hulladéklerakó és közvetlen környezete került a tanítóadatok közé, illetve a Kiskörei víztároló is. A romániai hulladéklerakókat egy helyi weboldalon lehet megtalálni, a hozzájuk tartozó koordinátákkal együtt [11]. Az ott bemutatott 46 hulladéklerakó közül 29 volt alkalmas tanításra: sok hulladéklerakó be lett tömve, vagy föld alatt működik. Minden hulladéklerakóhoz letöltöttem egy-egy nyári+tavaszi, téli és őszi multispektrális műholdképet, melyeket kézzel annotáltam. A nyári és tavaszi képeket azért vontam egybe, mivel ezek hulladékdetektálás szempontjából hasonló adatokat eredményeztek. A tanítóadatok pixelenként vannak előállítva, így a végső adathalmaz 27 millió tanítóadatból (pixelből) áll. Minden pixelhez hozzá van rendelve a vörös, kék, zöld, közeli infravörös sáv, illetve a "PI", "NDWI", "NDVI", "RNDVI", "SR" indexek. Ezen felül minden pixel címkézve van a 4.1 táblázatban leírtak szerint.

Címke azonosító	Címke neve	Címke magyarázat
100	Hulladék	Azon területek, melyeken hulladékot találtunk.
200	Víz	olyan területek, melyeken kizárólag vizet találtunk, általában folyók.
300	Legelők/Erdők	Zöld övezetből álló vad területek. Ezek lehetnek fák lombjai vagy füves zónák.
400	Mezők	Olyan földes területek, melyek meg vannak művelve, illetve ahol mezőgazdasági növények találhatók, például gabonafélék.
500	Ismeretlen	Olyan területek, melyek a korábbi kategóriákba nem sorolhatók bele. Ilyenek az épületek, aszfaltozott utak, háztetők, mezei utak.

4.1. táblázat. A tanítóadatok címkéi

4.0.2. Tanítási paraméterek

A nagy adathalmaz miatt a Random Forest modell is nagyon nagy lesz (körülbelül 14GB), ami egy nehezen kezelhető méret, így érdemes módosítani a modell paraméterein, hogy ez kisebb méretű legyen. A legjobb eredményeket azzal értem el, hogy a Random Forest fák méretét 20 mélységűre limitáltam. Ennek köszönhetően a modellek méretét 2GB-ra tudtam csökkenteni, és a ?? ábrából látható, hogy a csökkentett modell is hasonlóan teljesít a nagy modellhez képest.

Továbbiakban felmerült az a probléma is, hogy a tanítóadatok nagyon aránytalanok voltak: A 4.1 ábrából látható, hogy nagyságrendekkel kevesebb adattal rendelkezünk hulladékról, mint az összes többi adatról. Emiatt a modell nagyon sok false-negatívot termelt. Ennek korrigálására súlyokat alkalmaztam a tanítóadatokra. A súlyok kiszámolásához az összes címkére a 4.1 képletet használtam.

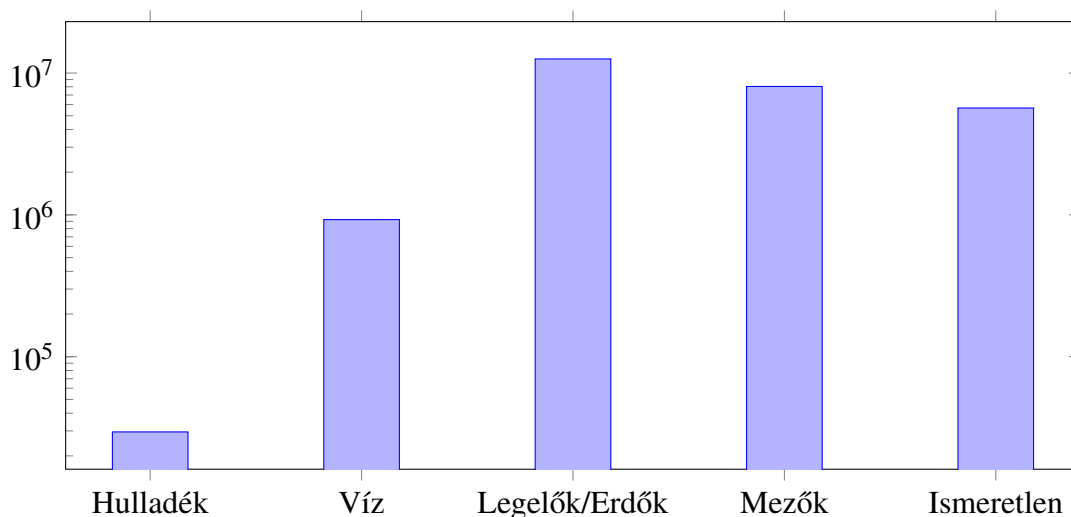
$$\text{címke súly} = \frac{\text{adathalmaz mérete}}{\text{címke darabszáma}} \quad (4.1)$$

4.0.3. Főkomponens analízis(PCA)

A modell méretének a csökkentésére még megpróbáltam a főkomponens analízis (PCA) alkalmazását is [12]. A módszer többek között arra is használható, hogy egy többdimenziós adathalmazból kivonja a legfontosabb információkat egy alacsonyabb dimenziószámú adathalmazba. Az ötlet az volt, hogy a bemeneti adatok dimenziószámának a csökkentésével csökkenni fog a modell mérete, de érdekes módon

táblázat a fák méretéről, a modellek méretéről és a különböző mélységekről

ábra készítés ide

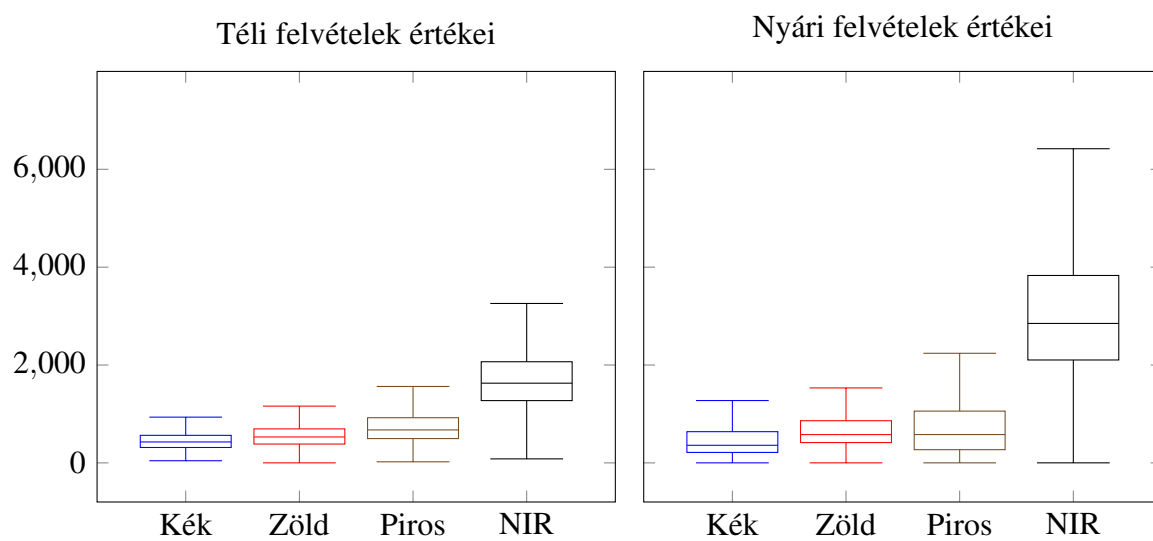


4.1. ábra. Az adatok közötti aránytalanság, logaritmikus skálázással

a modell mérete nem csökkent a dimenziószám csökkentésével, helyette lényegesen megnőtt. További vizsgálatok után kiderült, hogy hogyha kevesebb dimenziójú adatot adtam a modellnek, akkor a mérete lényegesen megnőtt.

4.0.4. Nyári és téli adatokra való lebontás

Alapértelmezetten a nyári és téli adatok között lényeges különbség tud lenni távérzékelés szempontból Közép-Európa területén: a téli időszakokban gyéresebb a vegetáció, ködösebb a levegő, illetve a nap sem süt ugyanabból a szögéből. Ez befolyásolhatja a modell pontosságát is az adott időszakokban. A nyári időszak alatt a márciustól októberig tartó időszakra gondolok, és a téli időszak alatt a novembertől februárig tartó időszakra gondolok. A 4.2 ábrából látható, hogy főleg a közeli infravörös (NIR) sávokon nagy eltérések vannak a nyári és téli felvételek között. Ennek fényében betanítottam külön egy nyári és egy téli modellt, melyek teljesítményét a ?? fejezetben részletezem.



4.2. ábra. Nyári és téli adatok összehasonlítása

5. fejezet

Tesztelés

5.0.1. Mérési módszerek

A teszteléshez előállítottam egy teszhalmazt, amivel a modell eredményeit összehasonlítottam. Az eredményeket a "Confusion Matrix" módszerével értékeltem ki [13].

6. fejezet

Megvalósítás és alkalmazás

6.0.1. A meglevő alkalmazások bővítése

Az asztali alkalmazás bővítése

A meglevő asztali alkalmazás alkalmas volt a tanítóadatok hatékony előállítására, de utólag nem lehetett visszanézni, hogy adott műholdfelvételhez milyen tanítóadatok tartoznak, illetve azt sem, hogy az adott tanítóadat hol volt mintavételezve. Az alkalmazás eredetileg egy CSV fájlban [14] tárolta el az összes pixel spektrális értékeit és indexeit, és ezt lehetett használni tanításra. Ennek az volt a hátránya, hogy nehéz volt áttekinteni illetve kiegészíteni az adatokat. Ezért az asztali alkalmazást kiegészítettem ezzel a funkcionalitással, a tanítóadatok előállítása elmentésekor az alkalmazás létrehoz egy külön raszteres réteget is külön minden műholdfelvételhez, melyen látható, hogy mely területek voltak hozzáadva a tanítóadatok közé, így tetszőleges módon előállítható/ellenőrizhető a tanítóhalmaz.

A szerveralkalmazás bővítése

A szerveralkalmazás és webalkalmazás is bővítésre került: a szerveralkalmazás mostmár több modellt is le tud futtatni a letöltött műholdfelvételeken és ezeket külön tárolja. A webalkalmazás mostmár képes letölteni külön ezeket az eredményeket és több hulladékmaszkoló módszer eredményét is meg tudja jeleníteni, ennek köszönhetően ezeket egymással össze lehet könnyen hasonlítani valós tesztadatokon.

6.0.2. A Tiszta-Tisza alkalmazás

A Tiszta-Tisza webalkalmazás a PET Kupa által használt webalkalmazás, melynek az

melyik link
kerüljön ide?

a célja, hogy egy olyan felületet biztosítson, ahol meglehetősen tekinteni a jelenleg ismert folyómentén található hulladéklerakókat, illetve akár a regisztrált felhasználók is be tudnak jelenteni ilyet. A PET Kupa megbízta az egyetemet azzal a feladattal, hogy ezt továbbfejlessze, és a feladatok közé tartozott az is, hogy a Random Forest modell eredményeit integráljuk ebbe az alkalmazásba. Ezt a feladatot én vállaltam el.

Tekintve arra, hogy a Tisza-Tisza térképén pontok vannak megjelenítve, a modell által detektált területeket is pontokkal jelöljük. Ehhez egy nagyobb terület közepére helyezünk el egy pontot. Előfordulhat olyan is, hogy a modell olyan képeket klasszifikál, melyek el vannak torzítva (például magas páratartalom miatt). Ilyenkor a false-positive-ok aránya lényegesen megnő. Ennek korrigálására a Tisza-Tisza alkalmazásban a legutolsó három detektálást (legfeljebb 1 hónap különbséggel) veszem figyelembe és két kép közös metszetével döntöm el, hogy milyen területek kerülnek fel a térképre. A lépéseket a 6.0.3 fejezetben részletezem.

A képfeldolgozás gyorsítása

A meglevő képfeldolgozó algoritmuson gyorsítani kellett, hogy elfogadható időn belül tudja feldolgozni a tanítóadatokat, illetve a naponta letöltött műholdfelvételeket a szerveren. Ezt a kutatólabor korábbi cikkjében párhuzamosítással javasolták, de egy egyszerű megoldással lényegesen tudtam gyorsítani a képfeldolgozást: Ehhez egy nagyon hatékony Python programcsomagot, a Numpy-t [15] használtam fel, mellyel lényegesen megnöveltem a feldolgozás sebességét: A tanítóadatok feldolgozásakor a régi módszer (6.1 forráskód) 16 felvételt tudott feldolgozni 4 nap és 10 óra alatt, míg az átírt módszer (6.2 forráskód) feldolgozott 85 felvételt 20 perc alatt. Az átírás technikája az ábrából látható.

```
1 def _calculate_index(numerator: np.ndarray, denominator: np.ndarray
2     ) -> np.ndarray:
3     """
4     Calculating an index based on given numerator and denominator.
5
6     :param numerator: numerator matrix
7     :param denominator: denominator matrix
8     :return: result matrix, containing the calculated values
9     """
10    # variables
```



```

11     rows = numerator.shape[0]
12     cols = numerator.shape[1]
13     index = np.ndarray(
14         shape=numerator.shape,
15         dtype="float32",
16     )
17
18     # calculate index
19     for i in range(rows):
20         for j in range(cols):
21             if np.isnan(numerator[i, j]) or np.isnan(denominator[i,
22                 j]):
23                 index[i, j] = float("NaN")
24             elif denominator[i, j] != 0:
25                 index[i, j] = numerator[i, j] / denominator[i, j]
26             else:
27                 if numerator[i, j] < 0:
28                     index[i, j] = np.nanmin(numerator)
29                 elif numerator[i, j] > 0:
30                     index[i, j] = np.nanmax(numerator)
31                 else:
32                     index[i, j] = float("NaN")
33
34     # return index values
35     return index

```

6.1. forráskód. A képfeldolgozás régi módszere

```

1 def calculate_index(numerator: np.ndarray, denominator: np.ndarray)
2     -> np.ndarray:
3     """
4     Calculating an index based on given numerator and denominator.
5
6     :param numerator: numerator matrix
7     :param denominator: denominator matrix
8     :return: result matrix, containing the calculated values
9     """
10
11     # variables
12     index = np.ndarray(
13         shape=numerator.shape,

```

```
13     dtype="float32",
14 )
15
16     numerator_nan_min = np.nanmin(numerator)
17     numerator_nan_max = np.nanmax(numerator)
18
19     # calculate index
20     nan_mask = np.isnan(numerator) | np.isnan(denominator)
21     numerator_zero_mask = numerator == 0
22     denominator_zero_mask = denominator == 0
23
24     invalid_mask = nan_mask | (numerator_zero_mask &
25                               denominator_zero_mask)
26     valid_mask = np.logical_not(invalid_mask)
27
28     valid_denominator_non_zero_mask = valid_mask & np.logical_not(
29         denominator_zero_mask)
30     valid_denominator_zero_mask = valid_mask &
31         denominator_zero_mask
32
33     numerator_positive_denominator_zero_mask =
34         valid_denominator_zero_mask & (numerator > 0)
35     numerator_negative_denominator_zero_mask =
36         valid_denominator_zero_mask & (numerator < 0)
37
38     index[invalid_mask] = float("NaN")
39     index[numerator_positive_denominator_zero_mask] =
40         numerator_nan_max
41     index[numerator_negative_denominator_zero_mask] =
42         numerator_nan_min
43     index[valid_denominator_non_zero_mask] = (
44         numerator[valid_denominator_non_zero_mask] / denominator[
45             valid_denominator_non_zero_mask]
46     )
47
48     # return index values
49     return index
```

6.2. forráskód. A képfeldolgozás numpy-al

6.0.3. Közös metszet

A már meglevő szervertől poligonok formájában, GeoJSON-ben [16] lehet lekérni az adott napon detektált hulladékterületeket. Így érdemes poligonok metszetében kigondolni a többségi szavazást. Jelöljük $BUF(P,n)$ -vel egy multipoligon puffert, ahol

$$P \in \mathbb{P}$$

egy multipoligon, és n egy egész szám. Ekkor a többségi szavazást három képre a 6.1 képlet szerint lehet alkalmazni. Ezután az elég nagy poligonok egy-egy belső pontját megválasztva megtudjuk jelölni a hulladéklerakókat.

$$\text{Eredmény multipoligon} = \bigcup_{P_1 \in \mathbb{P}} \bigcup_{P_2 \in \mathbb{P}} BUF(BUF(P_1, n) \cap BUF(P_2, n), -n) \quad (6.1)$$

hasonló
képleteket
mások is
ismernek
fórumokban,
de hivatalos
forrással nem
találkoztam

7. fejezet

Összefoglalás és eredmények

Köszönetnyilvánítás

Amennyiben a TDK projektet pénzügyi támogatást kapott egy projektből vagy az egyetemtől, jellemzően kötelező feltüntetni a dolgozatban is. A dolgozat elkészítéséhez segítséget nyújtó oktatók, hallgatótársak, kollégák felé is nyilvánítható külön köszönet.

A. függelék

Szimulációs eredmények

Irodalomjegyzék

- [1] M.G. Kibria, N.I. Masuk és R. et al. Safayet. „Plastic Waste: Challenges and Opportunities to Mitigate Pollution and Effective Management”. *International Journal of Environmental Research* 17.20 (2023. jan.). ISSN: 2008-2034. URL: <https://doi.org/10.1007/s41742-023-00507-z>.
- [2] Björn Nyberg és tsai. „Leaving a plastic legacy: Current and future scenarios for mismanaged plastic waste in rivers”. *Science of The Total Environment* 869 (2023), 161821. old. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2023.161821>. URL: <https://www.sciencedirect.com/science/article/pii/S0048969723004369>.
- [3] Tim H. M. van Emmerik és tsai. „River plastic transport and deposition amplified by extreme flood”. *Nature Water* 1.6 (2023), 514–522. old. ISSN: 2731-6084. DOI: [10.1038/s44221-023-00092-7](https://doi.org/10.1038/s44221-023-00092-7). URL: <https://doi.org/10.1038/s44221-023-00092-7>.
- [4] Leo Breiman. „Random Forests”. *Machine Learning* 45.1 (2001), 5–32. old. ISSN: 1573-0565. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). URL: <https://doi.org/10.1023/A:1010933404324>.
- [5] Dávid Magyar és tsai. „Waste Detection and Change Analysis based on Multispectral Satellite Imagery”. 2023. jan. DOI: [10.48550/arXiv.2303.14521](https://arxiv.org/abs/10.48550/arXiv.2303.14521).
- [6] Anjar Dimara Sakti és tsai. „Identification of illegally dumped plastic waste in a highly polluted river in Indonesia using Sentinel-2 satellite imagery”. *Scientific Reports* 13.1 (2023), 5039. old. ISSN: 2045-2322. DOI: [10.1038/s41598-023-32087-5](https://doi.org/10.1038/s41598-023-32087-5). URL: <https://doi.org/10.1038/s41598-023-32087-5>.
- [7] Gil Gonçalves és Umberto Andriolo. „Operational use of multispectral images for macro-litter mapping and categorization by Unmanned Aerial Vehicle”. *Marine Pollution Bulletin* 176 (2022), 113431. old. ISSN: 0025-326X. DOI: <https://doi.org/10.1016/j.marpolbul.2022.113431>.

- doi . org / 10 . 1016 / j . marpolbul . 2022 . 113431. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X22001138>.
- [8] Antonio Lanorte és tsai. „Agricultural plastic waste spatial estimation by Landsat 8 satellite images”. *Computers and Electronics in Agriculture* 141 (2017), 35–45. old. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2017.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169917302259>.
- [9] Planet.com. URL: <https://developers.planet.com/docs/apis/data/sensors/> (elérés dátuma 2024. 04. 03.).
- [10] Planet.com. URL: <https://developers.planet.com/docs/apis/data/sensors/> (elérés dátuma 2024. 04. 03.).
- [11] InfoCons.ro. URL: <https://fiiunexemplu.ro/in-romania-exista-46-depozite-de-deseuri-gropi-de-gunoi/> (elérés dátuma 2024. 04. 02.).
- [12] Hervé Abdi és Lynne J. Williams. „Principal component analysis”. *WIREs Computational Statistics* 2.4 (2010), 433–459. old. DOI: <https://doi.org/10.1002/wics.101>. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.101>. URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101>.
- [13] Russell G. Congalton. „A review of assessing the accuracy of classifications of remotely sensed data”. *Remote Sensing of Environment* 37.1 (1991), 35–46. old. ISSN: 0034-4257. DOI: [https://doi.org/10.1016/0034-4257\(91\)90048-B](https://doi.org/10.1016/0034-4257(91)90048-B). URL: <https://www.sciencedirect.com/science/article/pii/S003442579190048B>.
- [14] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180. 2005. okt. DOI: 10.17487/RFC4180. URL: <https://www.rfc-editor.org/info/rfc4180>.
- [15] Charles R. Harris és tsai. „Array programming with NumPy”. *Nature* 585.7825 (2020. szept.), 357–362. old. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [16] H. Butler és tsai. *The GeoJSON Format*. RFC 7946. 2016. aug. DOI: 10.17487/RFC7946. URL: <https://www.rfc-editor.org/info/rfc7946>.

Ábrák jegyzéke

4.1. Az adatok közötti aránytalanság, logaritmikus skálázással	10
4.2. Nyári és téli adatok összehasonlítása	11

Táblázatok jegyzéke

4.1. A tanítóadatok címkéi	9
--------------------------------------	---

Algoritmusjegyzék

Forráskódjegyzék

6.1. A képfeldolgozás régi módszere	14
6.2. A képfeldolgozás numpy-al	15